

---

Subject: Re: Can I generate parameters for UMK?

Posted by [Sender Ghost](#) on Mon, 02 Mar 2015 10:48:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

sergeynikitin wrote on Mon, 02 March 2015 04:51 Can I write config (command-line parameters set) for UMK for compile in situation when path for MyApps every time is different?

I didn't find such command line option(s).

There is a possibility to create a different assemblies with these paths and use them instead.

The current version of umk searches assemblies and build methods in the following directories (in order):

`${HOME}/.upp/umk`

`${HOME}/.upp/theide`

`${HOME}/.upp/ide`

`${HOME}`

executable directory

So, it is possible to create (unique) assemblies (and/or build methods) in these directories.

sergeynikitin wrote on Mon, 02 March 2015 04:55 May be, We can generate Makefile based on umk calls, but not on gcc calls?

This is possible, if you want to use umk (or theide) to build different packages. But I think, this is out of scope of current version, which might be used to understand about possible command line options, related to building.

I can give you examples, which you might adapt to your needs. They build some packages from examples assembly.

The batch file example (build.bat, e.g. for Windows operating system(s)):

```
@echo off
```

```
setlocal
```

```
set CURDIR=%~dp0
```

```
set UPPDIR=C:\upp
```

```
set BUILDMETHOD=MSC9
```

```
set BUILDOPTIONS=-r
```

```
rem Empty to default package configuration
```

```
set MAINCONF=
```

```
set ASSEMBLY=examples
```

```
set PACKAGES=AddressBook;Puzzle;UWord
```

```
set OUTPUTDIR=%CURDIR%build
```

```
rem theide or umk
```

```
set BUILDER=theide
```

```
if exist "%OUTPUTDIR%" (rmdir /S /Q "%OUTPUTDIR%")
```

```
mkdir "%OUTPUTDIR%"
```

```
cd "%UPPDIR%"
```

```
for %%P in (%PACKAGES%) do (
```

```
%BUILDER% %ASSEMBLY% %%P %BUILDMETHOD% %BILDOPTIONS% %MAINCONF%
"%OUTPUTDIR%" || exit /B
)
```

The shell file example (build.sh, e.g. for Unix-like operating system(s)):

```
#!/bin/sh
```

```
CURDIR=${PWD}
BUILDMETHOD=GCC
BILDOPTIONS=-rs
# Empty to default package configuration
MAINCONF=
ASSEMBLY=examples
PACKAGES="AddressBook Puzzle UWord"
OUTPUTDIR=${CURDIR}/build
# theide or umk
BUILDER=theide
```

```
# Remove output directory
if [ -e "${OUTPUTDIR}" ]; then
    rm -rf "${OUTPUTDIR}"
fi;
```

```
# Create output directory with intermediate directories, if needed
mkdir -p "${OUTPUTDIR}"
```

```
for P in ${PACKAGES}; do
    ${BUILDER} ${ASSEMBLY} ${P} ${BUILDMETHOD} ${BILDOPTIONS} ${MAINCONF}
    "${OUTPUTDIR}" || break
done;
```

The Makefile example:

```
CURDIR= $(PWD)
BUILDMETHOD= GCC
BILDOPTIONS= -rs
# Empty to default package configuration
MAINCONF=
ASSEMBLY= examples
PACKAGES= AddressBook Puzzle UWord
OUTPUTDIR= $(CURDIR)/build
# theide or umk
BUILDER= theide
```

```
all: clean build
```

```
build: create_output_dir $(PACKAGES)
```

```
$(PACKAGES):  
  ${BUILDER} ${ASSEMBLY} $@ ${BUILDMETHOD} ${BUILDOPTIONS} ${MAINCONF}  
  "${OUTPUTDIR}"
```

```
create_output_dir:  
  mkdir -p "${OUTPUTDIR}"
```

```
clean:  
  rm -rf "${OUTPUTDIR}"
```

```
.PHONY: all
```

The more advanced build files need to create in case of build packages with different main configuration(s).

---